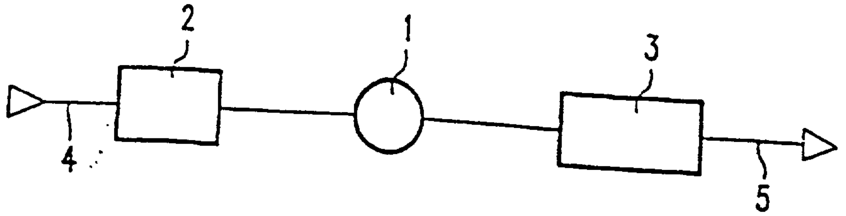


FIG. 1

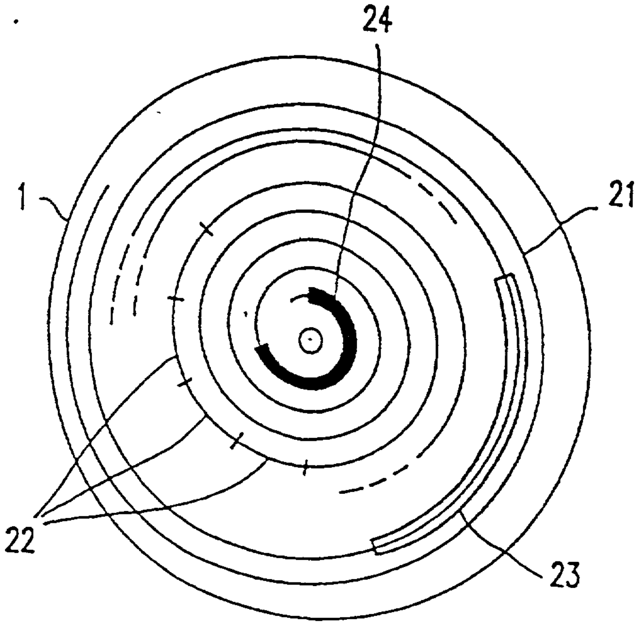


FIG. 2

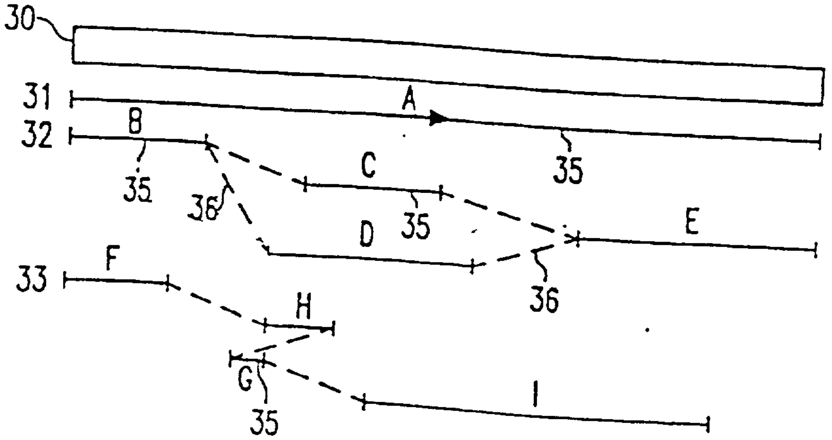


FIG. 3

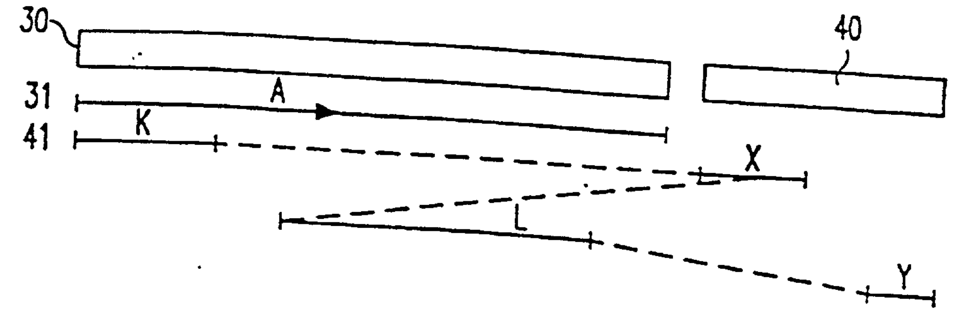

30
40
31
A
41
K
X
L
Y

FIG. 4

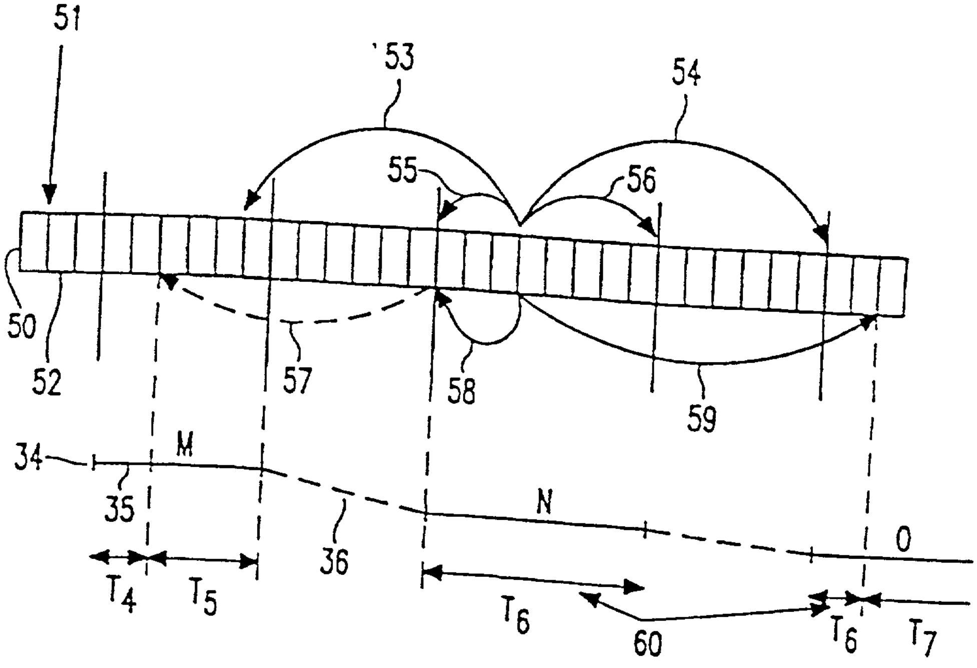

51
53
54
55
56
50
52
57
58
59
34
M
35
36
N
O
T_4
T_5
T_6
60
T_6
T_7

FIG. 5

```
path_descriptor
        descriptor_tag
        descriptor_length

        path_number

        PTS


        track_number
        pX_sectors_to_read
        pX_sectors_from_start
        pX_offset_next_section
        pX_offset_previous_section
        pX_offset_next_track
        pX_offset_previous_track
        pX_track_tc


    pX_path_tc
```

FIG. 6

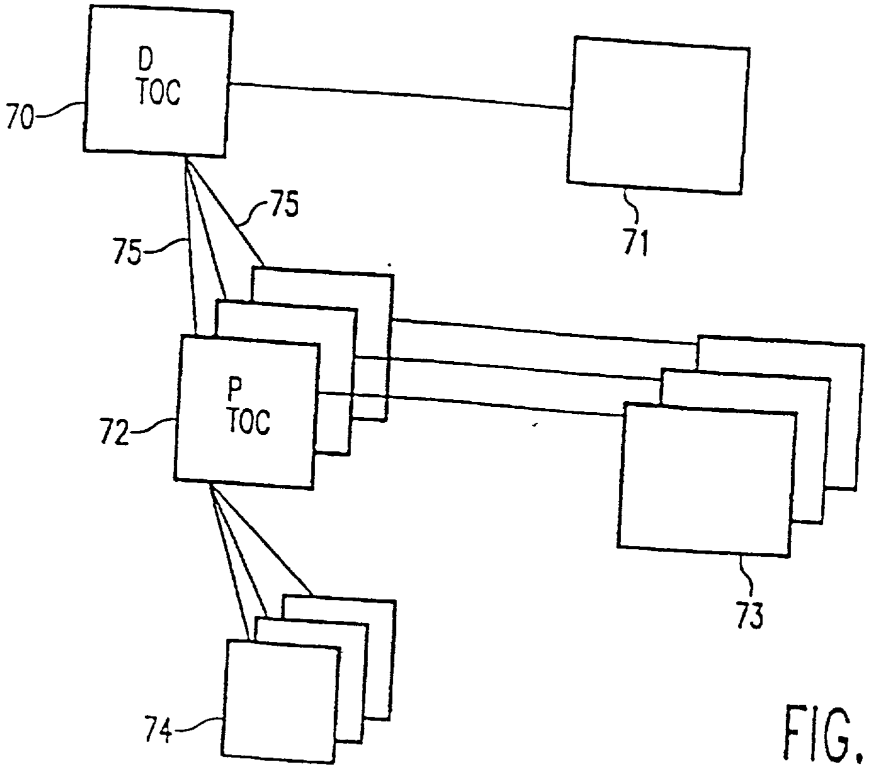


FIG. 7

Syntax	No. of bits	Mnemonic
path_descriptor()		
{		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
reserved	5	bslbf
path_number	3	uimsbf
reserved	4	bslbf
PTS[32..30]	3	bslbf
marker	1	bslbf
PTS[29..15]	15	bslbf
marker	1	bslbf
PTS[14..0]	15	bslbf
marker	1	bslbf
track_number	16	uimsbf
pX_sectors_to_read	24	uimsbf
pX_sectors_from_start	24	uimsbf
pX_offset_next_section	32	imsbf
pX_offset_previous_section	32	imsbf
pX_offset_next_track	32	imsbf
pX_offset_start_track	32	imsbf
pX_track_tc_hours_1	4	bslbf
pX_track_tc_hours_2	4	bslbf
pX_track_tc_minutes_1	4	bslbf
pX_track_tc_minutes_2	4	bslbf
pX_track_tc_seconds_1	4	bslbf
pX_track_tc_seconds_2	4	bslbf
pX_track_tc_CC	2	bslbf
pX_track_tc_frames_1	2	bslbf
pX_track_tc_frames_2	4	bslbf
pX_path_tc_hours_1	4	bslbf
pX_path_tc_hours_2	4	bslbf
pX_path_tc_minutes_1	4	bslbf
pX_path_tc_minutes_2	4	bslbf
pX_path_tc_seconds_1	4	bslbf
pX_path_tc_seconds_2	4	bslbf
pX_path_tc_CC	2	bslbf
pX_path_tc_frames_1	2	bslbf
pX_path_tc_frames_2	4	bslbf
}		

FIG. 6A

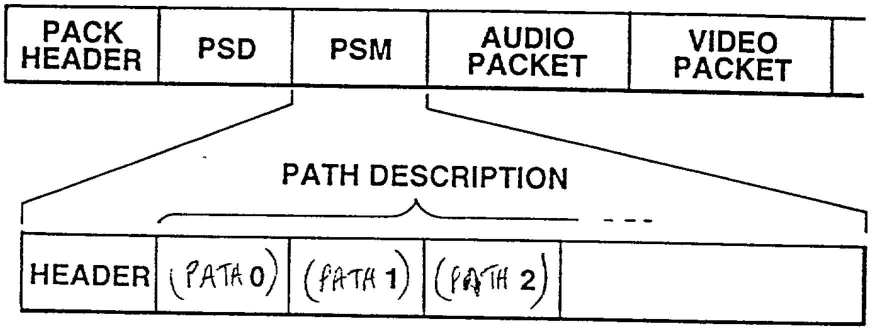

PACK HEADER
PSD
PSM
AUDIO PACKET
VIDEO PACKET
PATH DESCRIPTION
- - -
HEADER
(PATH 0)
(PATH 1)
(PATH 2)

FIG. 6B

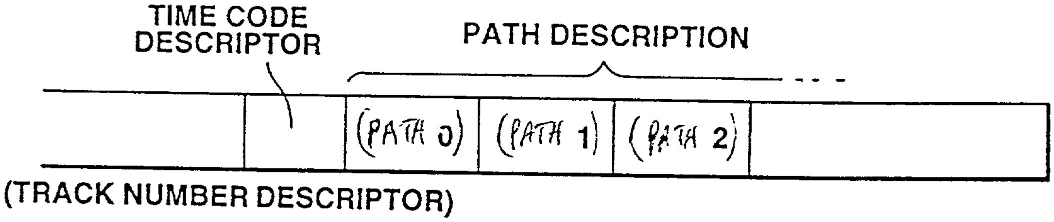

TIME CODE DESCRIPTOR
PATH DESCRIPTION
- - -
(PATH 0)
(PATH 1)
(PATH 2)
(TRACK NUMBER DESCRIPTOR)

FIG. 6C

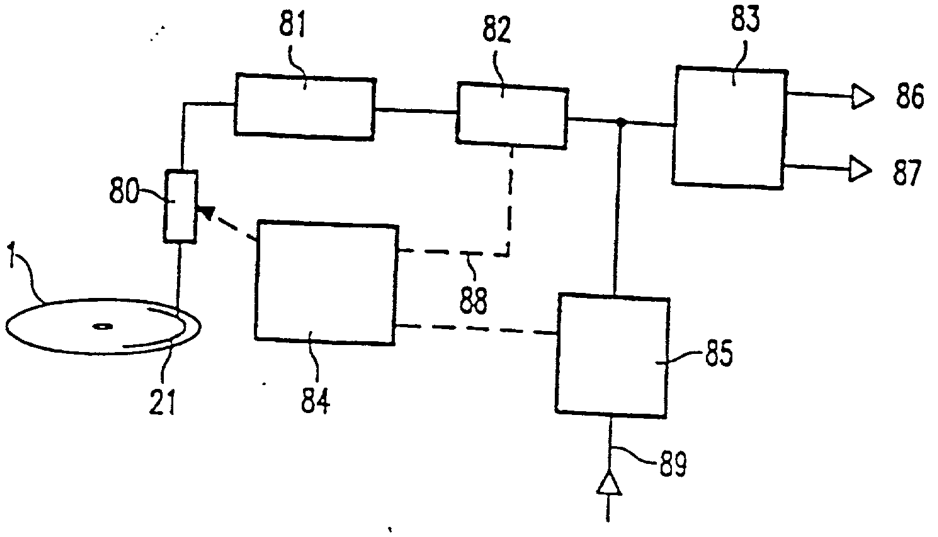


FIG. 8

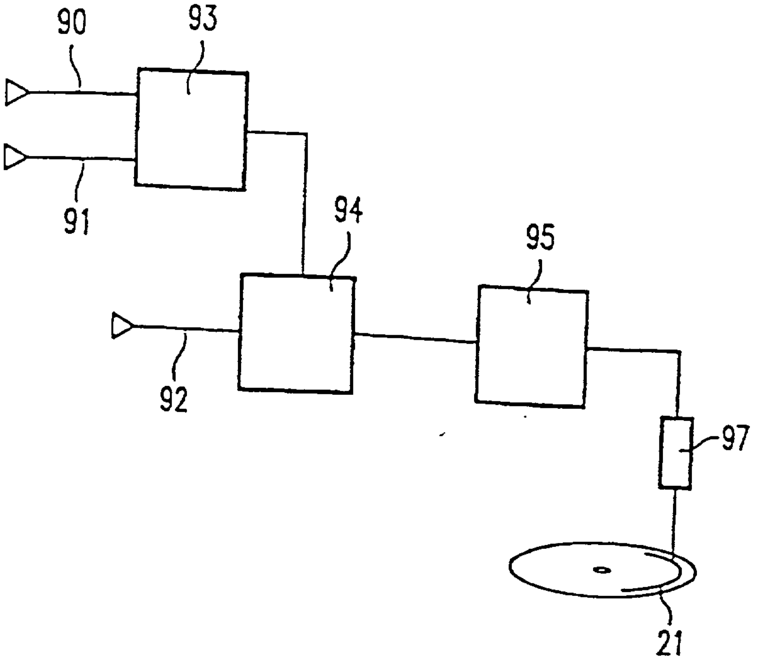


FIG. 9

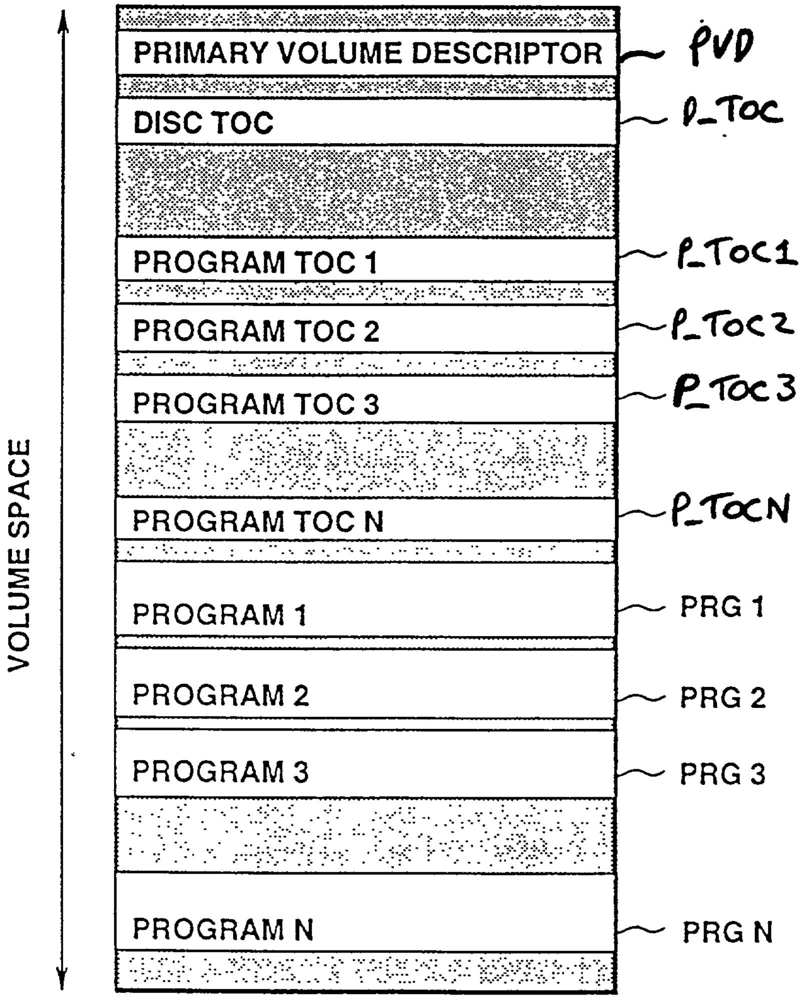

VOLUME SPACE
PRIMARY VOLUME DESCRIPTOR
PVD
DISC TOC
P_TOC
PROGRAM TOC 1
P_TOC1
PROGRAM TOC 2
P_TOC2
PROGRAM TOC 3
P_TOC3
PROGRAM TOC N
P_TOCN
PROGRAM 1
PRG 1
PROGRAM 2
PRG 2
PROGRAM 3
PRG 3
PROGRAM N
PRG N

FIG.10

BP	Field name	Content
1	Volume Descriptor Type	numerical value
2 to 6	Standard Identifier	CD001
7	Volume Descriptor Version	numerical value
8	Unused Field	(00) byte
9 to 40	System Identifier	a-characters
41 to 72	Volume Identifier	d-characters
73 to 80	Unused Field	(00) bytes
81 to 88	Volume Space Size	numerical value
89 to 120	Unused Field	(00) bytes
121 to 124	Volume Set Size	numerical value
125 to 128	Volume Sequence Number	numerical value
129 to 132	Logical Block Size	numerical value
133 to 140	Path Table Size	numerical value
141 to 144	Location of Occurrence of Type L Path Table	numerical value
145 to 148	Location of Optional Occurrence of Type L Path Table	numerical value
149 to 152	Location of Occurrence of Type M Path Table	numerical value
153 to 156	Location of Optional Occurrence of Type M Path Table	numerical value
157 to 190	Directory Record for Root Directory	34 bytes
191 to 318	Volume Set Identifier	d-characters
319 to 446	Publisher Identifier	a-characters
447 to 574	Data Preparer Identifier	a-characters
575 to 702	Application Identifier	a-characters
703 to 739	Copyright File Identifier	d-characters, SEPARATOR 1, SEPARATOR 2
740 to 776	Abstract File Identifier	d-characters, SEPARATOR 1, SEPARATOR 2
777 to 813	Bibliographic File Identifier	d-characters, SEPARATOR 1, SEPARATOR 2
814 to 830	Volume Creation Date and Time	Digit(s), numerical value
831 to 847	Volume Modification Date and Time	Digit(s), numerical value
848 to 864	Volume Expiration Date and Time	Digit(s), numerical value
865 to 881	Volume Effective Date and Time	Digit(s), numerical value
882	File Structure Version	numerical value
883	(Reserved for future standardization)	(00) byte
884 to 1 395	Application Use	not specified
1 396 to 2 048	(Reserved for future standardization)	(00) bytes

FIG. 11

Syntax	No. of Bits	Mnemonic
disc_toc (){		
dvd_signature	64	bslbf
dvd_version	32	bslbf
length	16	uimsbf
toc_type	8	bslbf
reserved	8	bslbf
album()		
catalogue()		
reserved	16	uimsbf
number_of_programs (NOP)	16	bslbf
rating_definitions_offset	32	uimsbf
program_linkage_offset	32	uimsbf
disc_track_offset	32	uimsbf
program_toc_pointers_offset	32	uimsbf
disc_play_time_offset	32	uimsbf
disc_name_offset	32	uimsbf
disc_date_offset	32	uimsbf
disc_copyright_offset	32	uimsbf
disc_publisher_offset	32	uimsbf
reserved	288	bslbf
rating_definitions()		
program_linkage()		
disc_tracks()		
program_toc_pointers()		
disc_play_time()		
disc_name()		
disc_date()		
disc_copyright()		
disc_publisher()		

FIG. 12

Syntax	No. of Bits	Mnemonic
disc_tracks() {		
for(i=0; i < 8; i++){		
number_of_tracks(NOT)	16	uimsbf
reserved	16	bslbf
for (t=0; t < NOT; t++) {		
program_number	16	uimsbf
track_number	16	uimsbf
start_lsa	32	uimsbf
last_lsa	32	uimsbf
last_es_lsa	32	uimsbf
stoppiing_stc	32	uimsbf
playing_time()		
}		
}		
}		

FIG. 12A

Syntax	No. of Bits	Mnemonic
program_toc(){		
dvd_signature	64	bslbf
dvd_version	32	bslbf
length	16	uimsbf
toc_type	8	bslbf
reserved	8	bslbf
number_of_tracks (NOT)	16	uimsbf
reserved	16	bslbf
program_start_lsa	32	uimsbf
path_rating_assignment_offset	32	uimsbf
for (i=0; i<8; i++){		
program_tracks_offset_i	32	uimsbf
}		
elementary_stream_info_offset	32	uimsbf
program_play_times_offset	32	uimsbf
program_name_offset	32	uimsbf
track_names_offset	32	uimsbf
program_date_offset	32	uimsbf
track_dates_offset	32	uimsbf
program_copyright_offset	32	uimsbf
program_publisher_offset	32	uimsbf
reserved	288	bslbf
path_rating_assignments()		
for (i=0; i<8; i++){		
if (program_chapters_offset_i>0){		
program_tracks()		
entry_points()		
path_table()		
}		
}		
elementary_stream_info()		
program_play_times()		
program_name()		
track_names()		
program_date()		
track_dates()		
program_copyright()		
program_publisher()		

FIG. 13

Syntax	No. of Bits	Mnemonic
program_tracks() {		
number_of_path_tracks(NOPT)	16	uimsbf
reserved	16	bslbf
for (t=0; t < NOPT; t++) {		
reserved	16	bslbf
track_number	16	uimsbf
start_rsa	32	uimsbf
last_rsa	32	uimsbf
last_es_rsa	32	uimsbf
stoppi ng_stc	32	uimsbf
track_playing_time()	32	uimsbf
}		
}		

FIG. 13A

Syntax	No. of Bits	Mnemonic
path_table() {		
number_of_sections(NOCH)	32	uimsbf
for (ch=0; ch < NOCH; ch++) {		
start_rsa	32	uimsbf
last_rsa	32	uimsbf
stoppi ng_stc	32	uimsbf
last_es_rsa	32	uimsbf
}		
}		

FIG. 13B

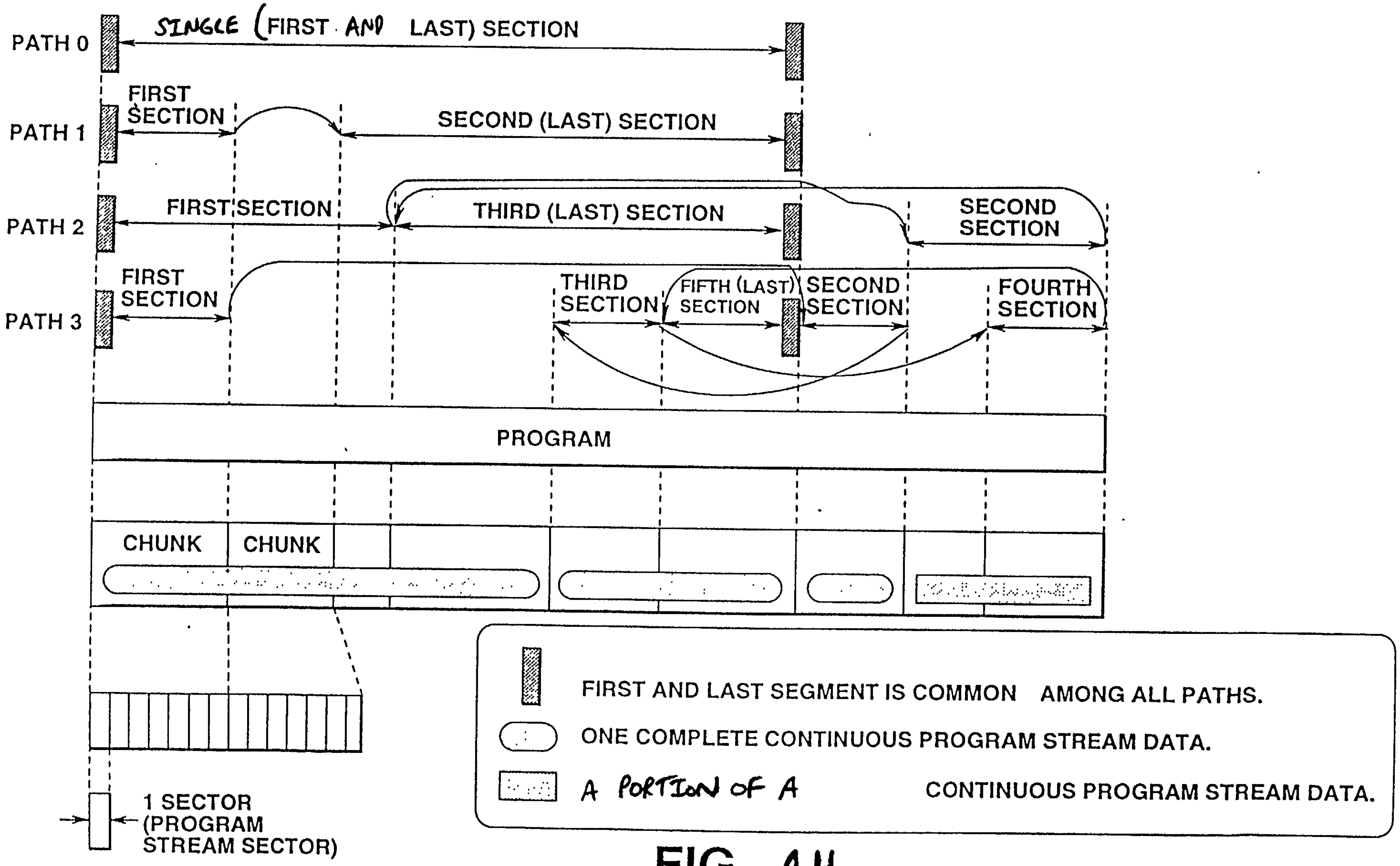

PATH 0
SINGLE (FIRST AND LAST) SECTION
PATH 1
FIRST SECTION
SECOND (LAST) SECTION
PATH 2
FIRST SECTION
THIRD (LAST) SECTION
SECOND SECTION
PATH 3
FIRST SECTION
THIRD SECTION
FIFTH (LAST) SECTION
SECOND SECTION
FOURTH SECTION
PROGRAM
CHUNK
CHUNK
1 SECTOR (PROGRAM STREAM SECTOR)
FIRST AND LAST SEGMENT IS COMMON AMONG ALL PATHS.
ONE COMPLETE CONTINUOUS PROGRAM STREAM DATA.
A PORTION OF A CONTINUOUS PROGRAM STREAM DATA.

FIG. 14

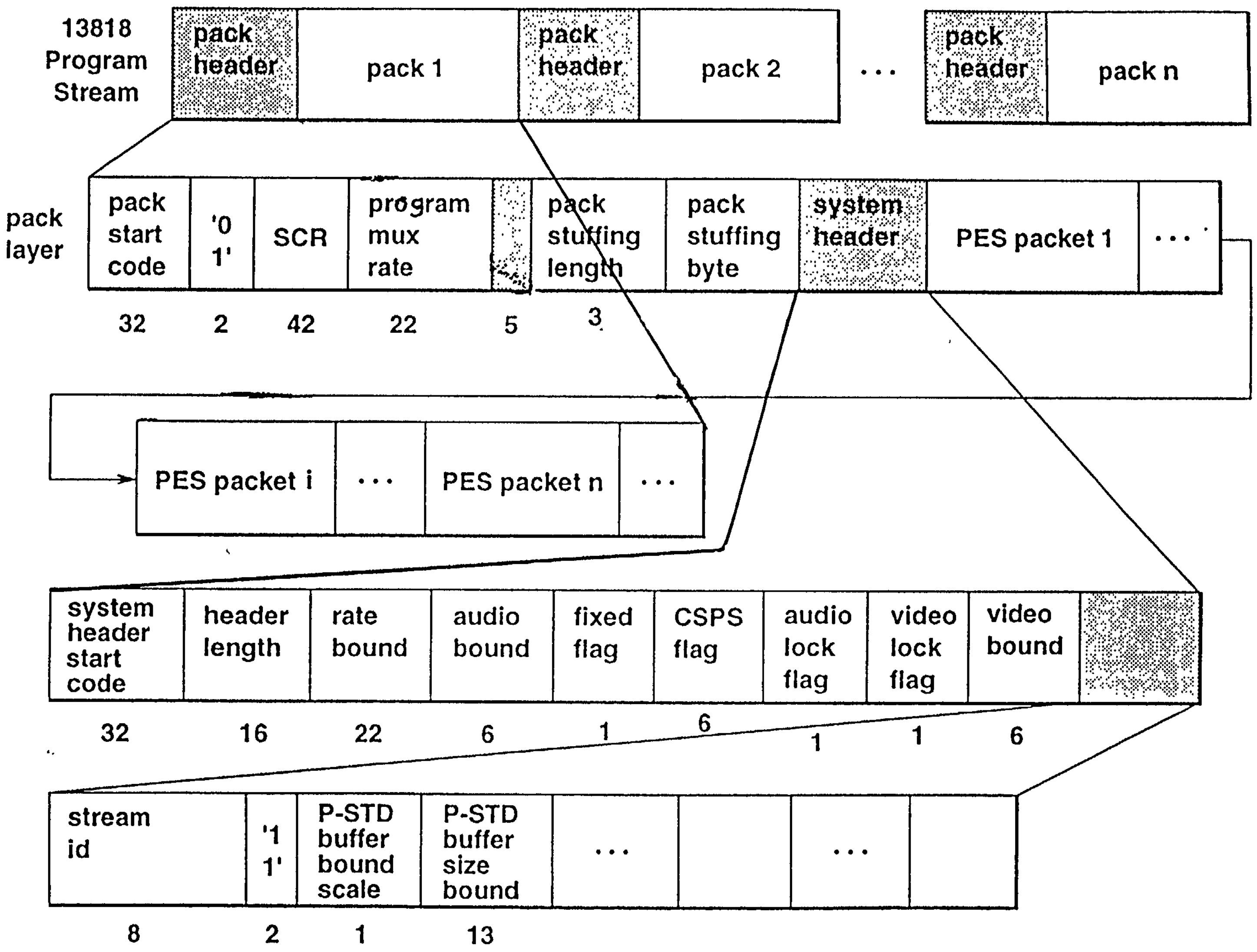

13818 Program Stream
pack header
pack 1
pack header
pack 2
. . .
pack header
pack n
pack layer
pack start code
'0 1'
SCR
program mux rate
pack stuffing length
pack stuffing byte
system header
PES packet 1
. . .
32
2
42
22
5
3
PES packet i
. . .
PES packet n
. . .
system header start code
header length
rate bound
audio bound
fixed flag
CSPS flag
audio lock flag
video lock flag
video bound
32
16
22
6
1
6
1
1
6
stream id
'1 1'
P-STD buffer bound scale
P-STD buffer size bound
. . .
. . .
8
2
1
13

FIG. 15A

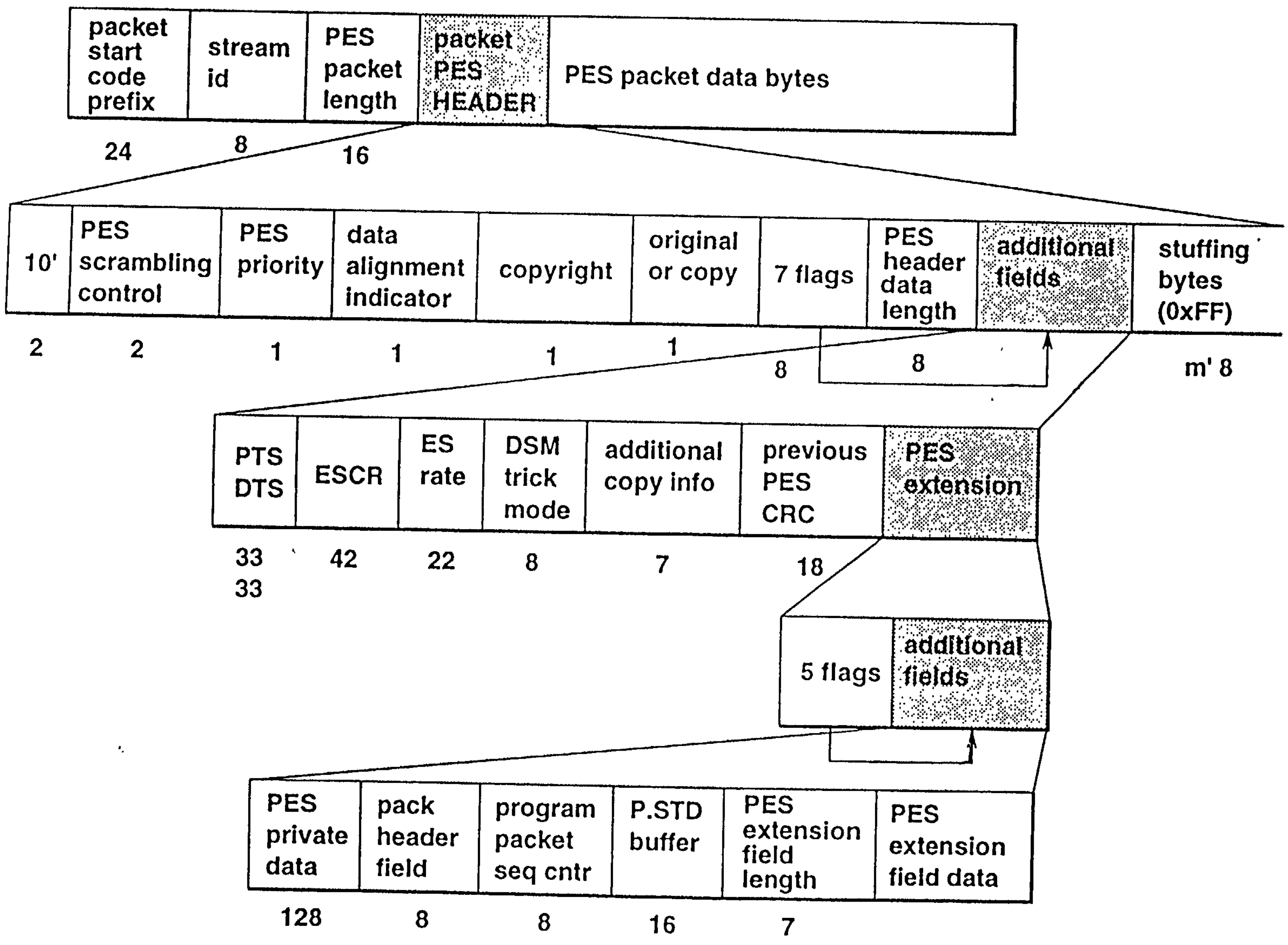

packet start code prefix
stream id
PES packet length
packet PES HEADER
PES packet data bytes
24
8
16
10'
PES scrambling control
PES priority
data alignment indicator
copyright
original or copy
7 flags
PES header data length
additional fields
stuffing bytes (0xFF)
2
2
1
1
1
1
8
8
m' 8
PTS DTS
ESCR
ES rate
DSM trick mode
additional copy info
previous PES CRC
PES extension
33 33
42
22
8
7
18
5 flags
additional fields
PES private data
pack header field
program packet seq cntr
P.STD buffer
PES extension field length
PES extension field data
128
8
8
16
7

FIG. 15B

Syntax	No. of Bits	Mnemonic
PES_packet(){		
packet_start_code_prefix	24	bslbf
stream_id	8	uimsbf
PES_packet_length	16	uimsbf
if (stream_id != program_stream_map		
&& stream_id !=padding_stream		
&& stream_id !=private_stream_2		
&& stream_id !=ECM		
&& stream_id !=EMM		
&& stream_id !=program_stream_directory		
&& stream_id !=DSMCC_stream		
&& stream_id !=ITU-T Rec. H.222.1 type E_stream){		
'10'	2	bslbf
PES_scrambling_control	2	bslbf
PES_priority	1	bslbf
data_alignment_indicator	1	bslbf
copyright	1	bslbf
original_or_copy	1	bslbf
PTS_DTS_flags	2	bslbf
ESCR_flag	1	bslbf
ES_rate_flag	1	bslbf
DSM_trick_mode_flag	1	bslbf
additional_copy_info_flag	1	bslbf
PES_CRC_flag	1	bslbf
PES_extension_flag	1	bslbf
PES_header_data_length	8	uimsbf
if (PTS_DTS_flag == '10') {		
'0010'	4	bslbf
PTS [32..30]	3	bslbf
marker_bit	1	bslbf
PTS [29..15]	15	bslbf
marker_bit	1	bslbf
PTS [14..0]	15	bslbf
marker_bit	1	bslbf
}		
if (PTS_DTS_flags == '11') {		
'0011'	4	bslbf
PTS [32..30]	3	bslbf
marker_bit	1	bslbf
PTS [29..15]	15	bslbf
marker_bit	1	bslbf
PTS [14..0]	15	bslbf
marker_bit	1	bslbf
'0001'	4	bslbf
DTS [32..30]	3	bslbf
marker_bit	1	bslbf
DTS [29..15]	15	bslbf

FIG. 16A

Syntax	No. of Bits	Mnemonic
marker_bit	1	bslbf
DTS [14..0]	15	bslbf
marker_bit	1	bslbf
}		
if (ESCR_flag == '1') {		
reserved	2	bslbf
ESCR_base [32..30]	3	bslbf
marker_bit	1	bslbf
ESCR_base [29..15]	15	bslbf
marker_bit	1	bslbf
ESCR_base [14..0]	15	bslbf
marker_bit	1	bslbf
ESCR_extension	9	uimsbf
marker_bit	1	bslbf
}		
if (ES_rate_flag == '1) {		
marker_bit	1	bslbf
ES_rate	22	uimsbf
marker_bit	1	bslbf
}		
if (DSM_trick_mode_flag == '1) {		
trick_mode_control	3	uimsbf
if (trick_mode_control == fast_forward) {		
field_id	2	bslbf
intra_slice_refresh	1	bslbf
frequency_truncation	2	bslbf
}		
else if (trick_mode_control == slow_motion) {		
rep-cntrl	5	uimsbf
}		
else if (trick_mode_control == freeze_frame) {		
field_id	2	uimsbf
reserved	3	bslbf
}		
else if (trick_mode_control == fast_reverse') {		
field_id	2	bslbf
intra_slice_refresh	1	bslbf
frequency_truncation	2	bslbf
else if (trick_mode_control == slow_reverse) {		
rep_cntrl	5	uimsbf
}		
else		
reserved	5	bslbf
}		
if (additional_copy_info_flag == '1') {		
marker_bit	1	bslbf
additional_copy_info	7	bslbf

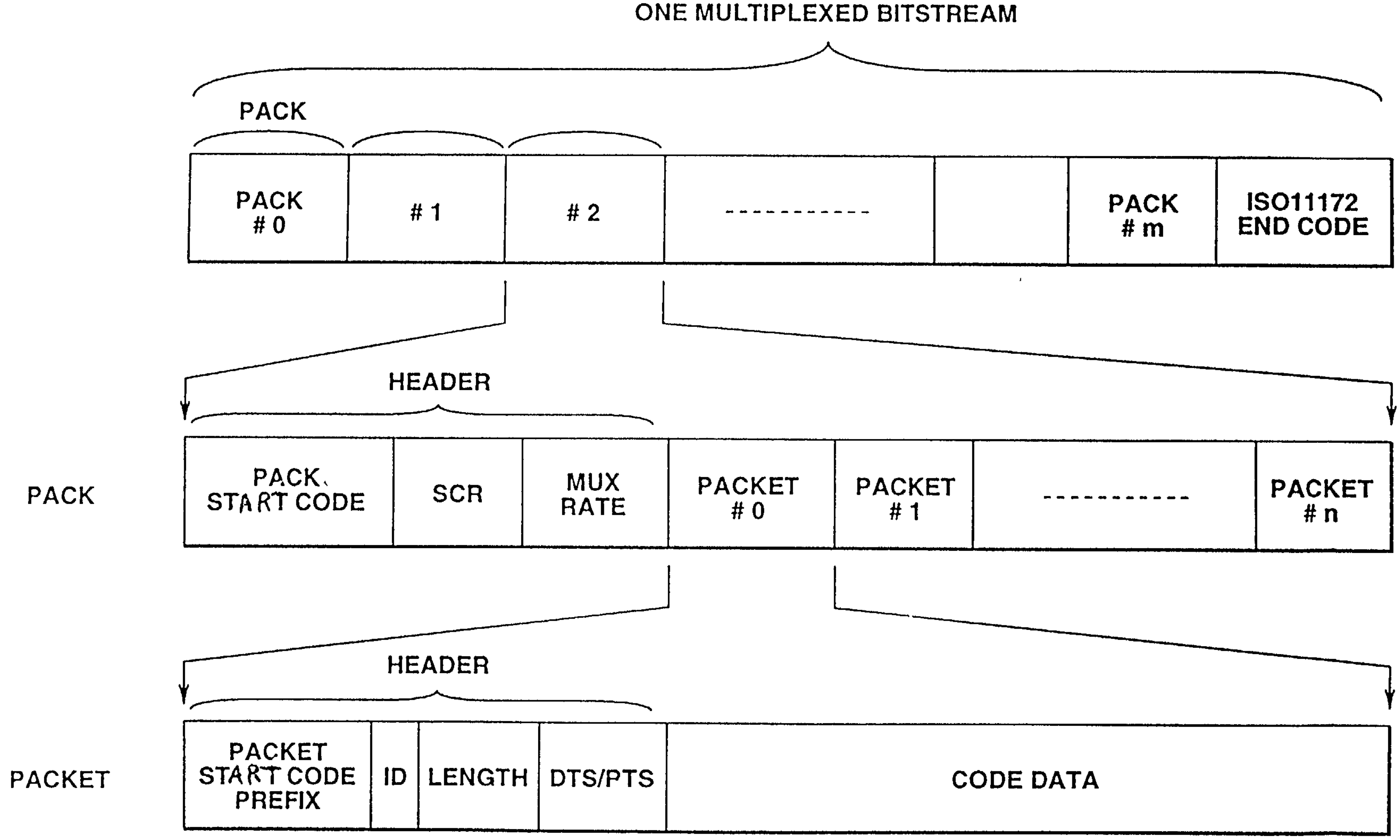
FIG. 16B

Syntax	No. of Bits	Mnemonic
}		
if (PES_CRC_flag == '1') {		
previous_PES_packet_CRC	16	bslbf
}		
if (PES_extension_flag == '1') {		
PES_private_data_flag	1	bslbf
pack_header_field_flag	1	bslbf
program_packet_sequence_counter_flag	1	bslbf
P-STD_buffer_flag	1	bslbf
reserved	3	bslbf
PES_extension_flag_2	1	bslbf
if (PES_private_data_flag == '1') {		
PES_private_data	128	bslbf
}		
if (pack_header_field_flag == '1') {		
pack_field_length	8	uimsbf
pack_header()		
}		
if (program_packet_sequence_counter_flag == '1') {		
marker_bit	1	bslbf
program_packet_sequence_counter	7	uimsbf
marker_bit	1	bslbf
MPEG1_MPEG2_identifier	1	bslbf
original_stuff_length	6	uimsbf
}		
if (P-STD_buffer_flag == '1') {		
'01'	2	bslbf
P-STD_buffer_scale	1	bslbf
P-STD_buffer_size	13	uimsbf
}		
if (PES_extension_flag_2 == '1) {		
marker_bit	1	bslbf
PES_extension_field_length	7	uimsbf
for (i=0; i<PES_extension_field_length; i++)		
{		
reserved	8	bslbf
}		
}		
}		
for (i=0; i<N1; i++) {		
stuffing_byte	8	bslbf
}		
for (i=0; i<N2; i++) {		
PES_packet_data_byte	8	bslbf

FIG.16C

```
Syntax                                                          No. of Bits   Mnemonic
            }
        }
        else if (stream_id == program_stream_map
        || stream_id == private_stream_2
        || stream_id == ECM
        || stream_id == EMM
        || stream_id == program_stream_directory
        || stream_id == DSMCC_stream)
        || stream_id == ITU-T Rec. H.222.1 type E stream {
            for (i=0; i<PES_packet_length; i++) {
                PES_packet_data_byte                                  8       bslbf
            }
        }
        else if (stream_id == padding_stream) {
            for (i=0; i<PES_packet_length; i++) {
                padding_byte                                          8       bslbf
            }
        }
}
```

FIG.16D


ONE MULTIPLEXED BITSTREAM
PACK
PACK # 0
# 1
# 2

PACK # m
ISO11172 END CODE
HEADER
PACK
PACK START CODE
SCR
MUX RATE
PACKET # 0
PACKET # 1

PACKET # n
HEADER
PACKET
PACKET START CODE PREFIX
ID
LENGTH
DTS/PTS
CODE DATA

FIG.17

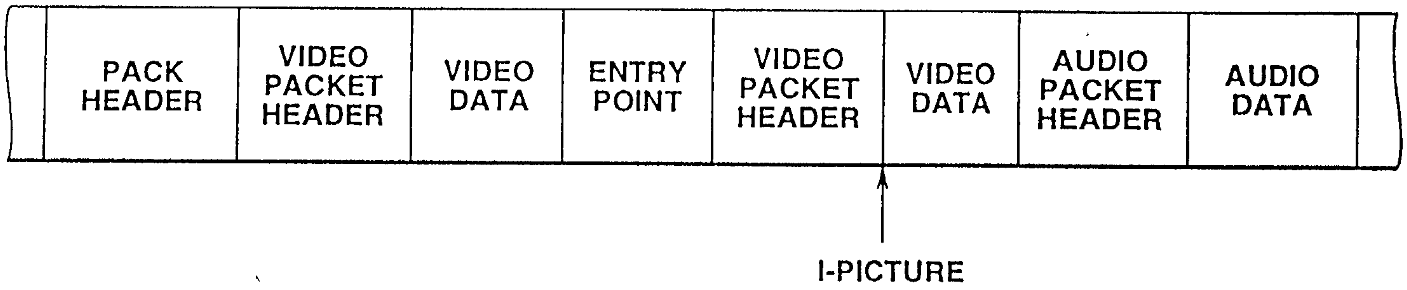

PACK HEADER
VIDEO PACKET HEADER
VIDEO DATA
ENTRY POINT
VIDEO PACKET HEADER
VIDEO DATA
AUDIO PACKET HEADER
AUDIO DATA
I-PICTURE

FIG.18

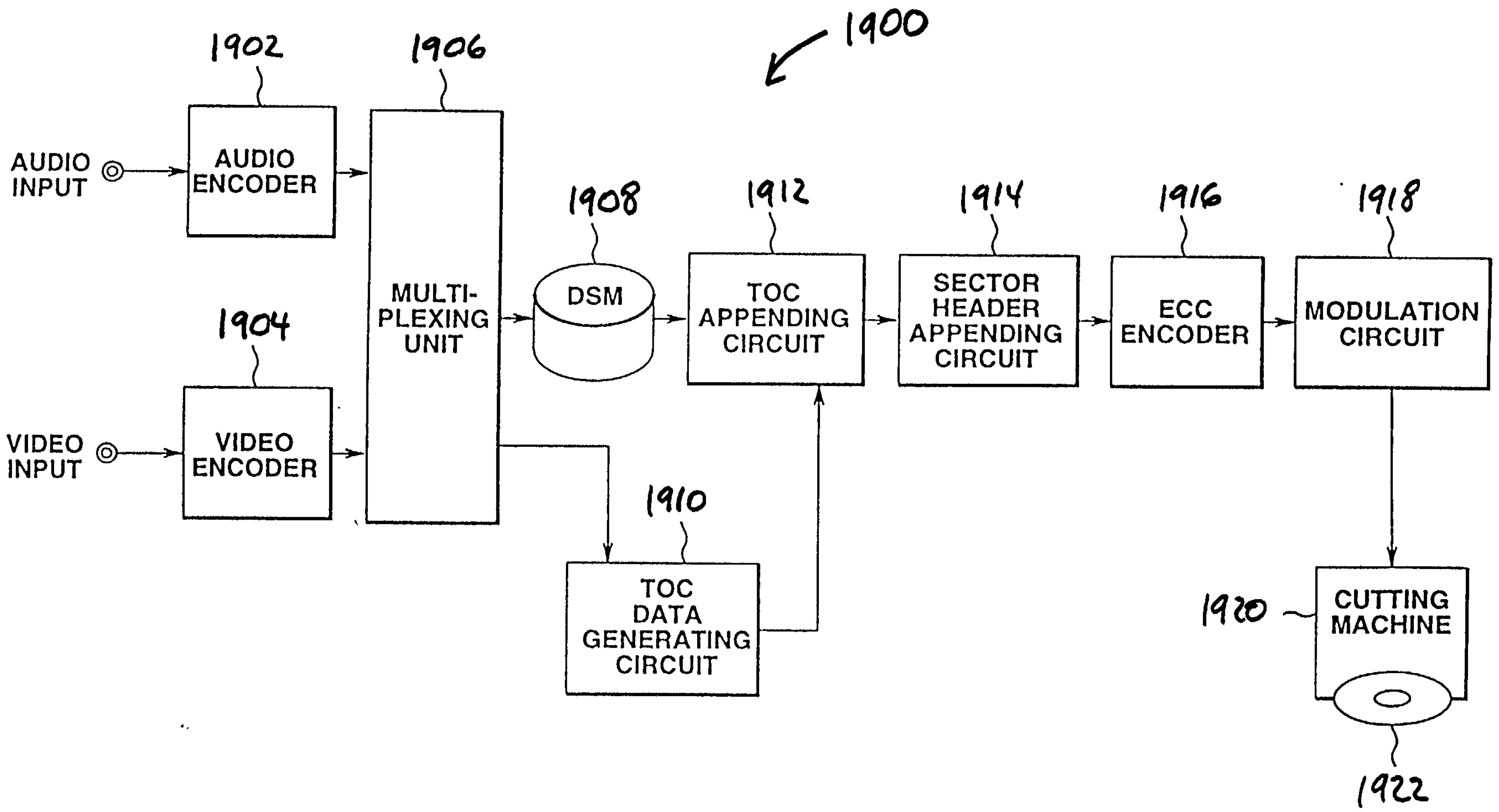

1900
1902
1904
1906
1908
1910
1912
1914
1916
1918
1920
1922
AUDIO INPUT
VIDEO INPUT
AUDIO ENCODER
VIDEO ENCODER
MULTI-PLEXING UNIT
DSM
TOC DATA GENERATING CIRCUIT
TOC APPENDING CIRCUIT
SECTOR HEADER APPENDING CIRCUIT
ECC ENCODER
MODULATION CIRCUIT
CUTTING MACHINE

FIG.19

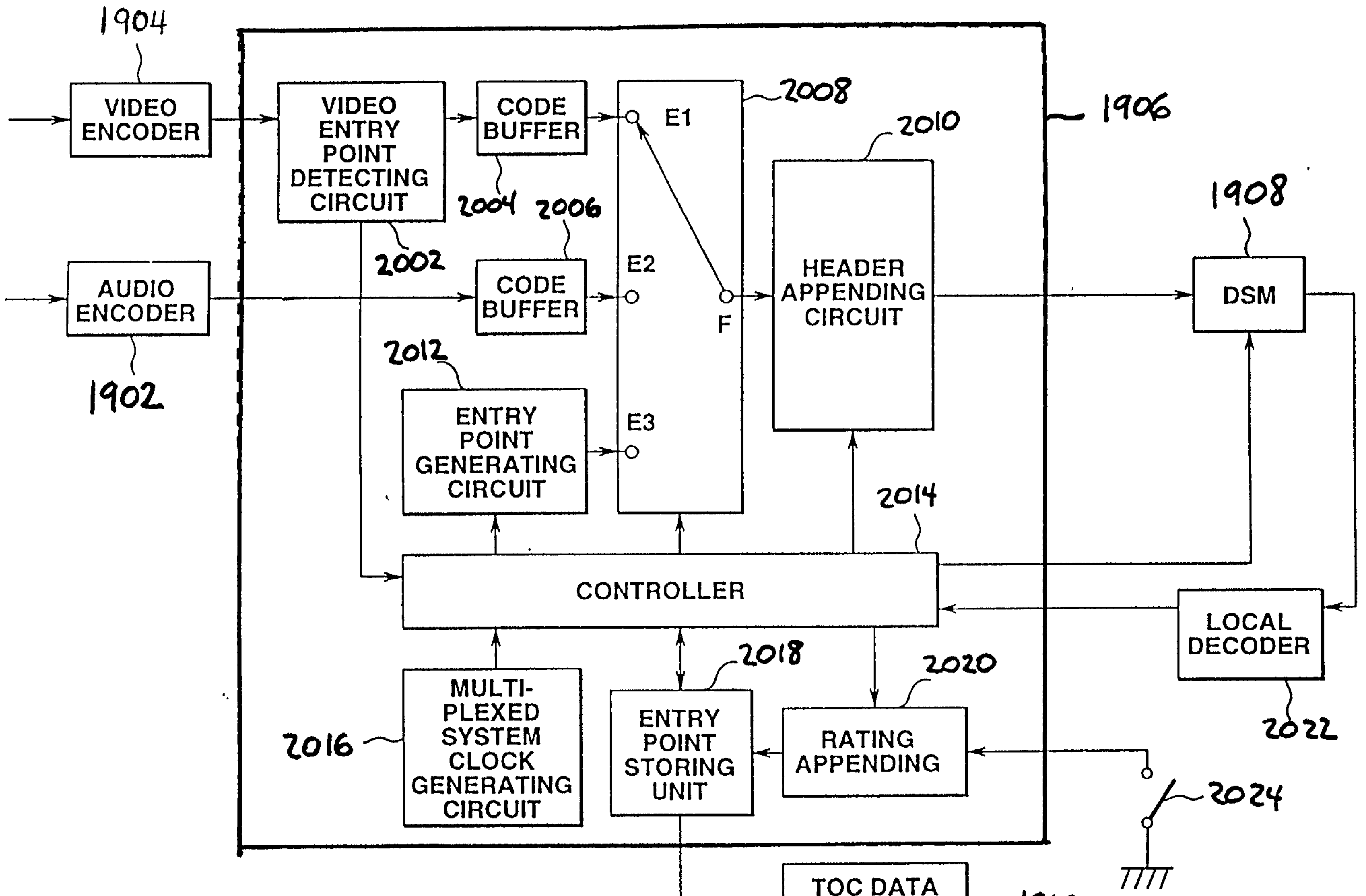

1904
VIDEO ENCODER
AUDIO ENCODER
1902
1906
VIDEO ENTRY POINT DETECTING CIRCUIT
2002
CODE BUFFER
2004
CODE BUFFER
2006
2012
ENTRY POINT GENERATING CIRCUIT
2008
E1
E2
E3
F
2010
HEADER APPENDING CIRCUIT
2014
CONTROLLER
2016
MULTI-PLEXED SYSTEM CLOCK GENERATING CIRCUIT
2018
ENTRY POINT STORING UNIT
2020
RATING APPENDING
1908
DSM
LOCAL DECODER
2022
2024
TOC DATA GENERATING CIRCUIT
1910

FIG. 20

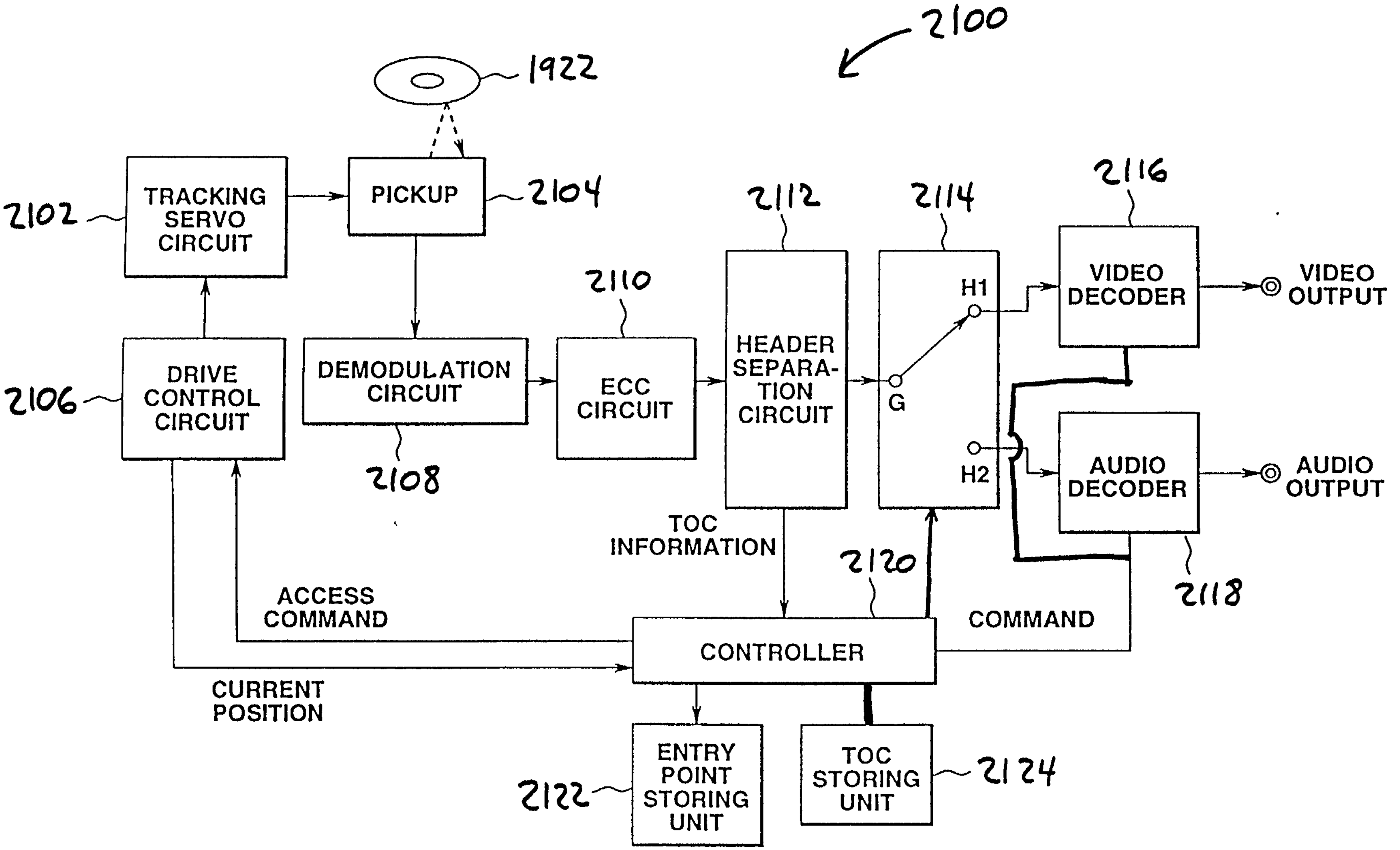

2100
1922
2102
TRACKING SERVO CIRCUIT
PICKUP
2104
2106
DRIVE CONTROL CIRCUIT
DEMODULATION CIRCUIT
2108
2110
ECC CIRCUIT
2112
HEADER SEPARA-TION CIRCUIT
2114
H1
G
H2
2116
VIDEO DECODER
VIDEO OUTPUT
AUDIO DECODER
AUDIO OUTPUT
2118
TOC INFORMATION
2120
ACCESS COMMAND
COMMAND
CONTROLLER
CURRENT POSITION
ENTRY POINT STORING UNIT
2122
TOC STORING UNIT
2124

FIG.21

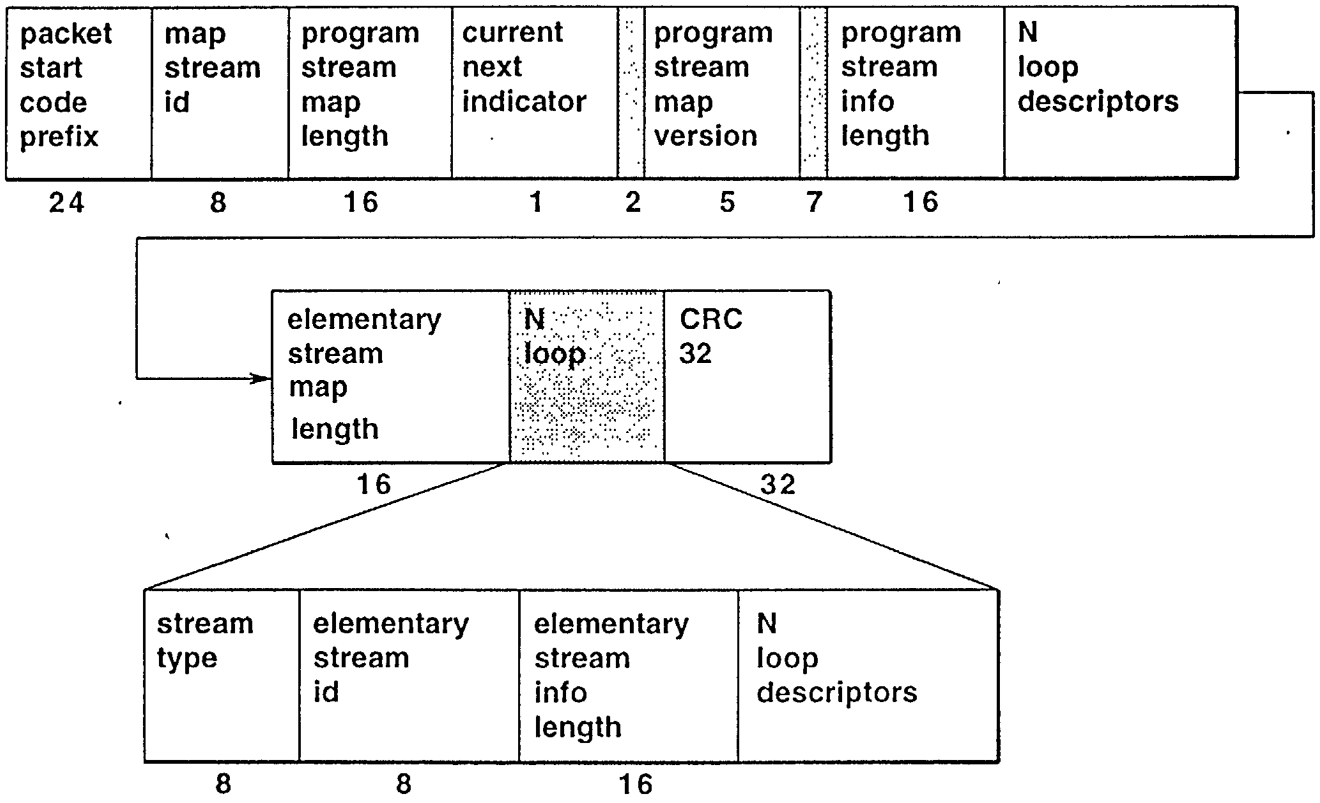
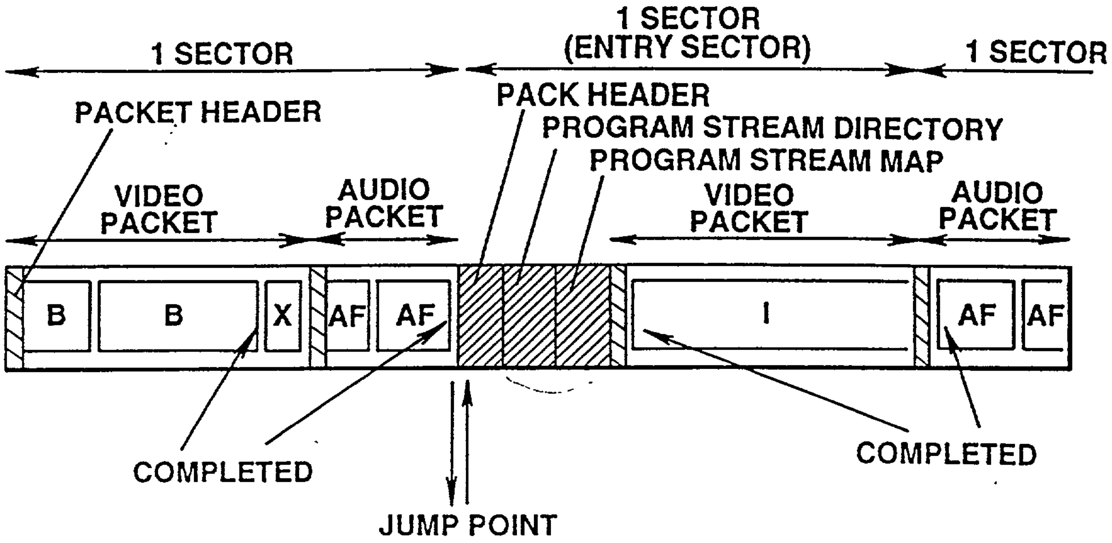

packet start code prefix
24
map stream id
8
program stream map length
16
current next indicator
1
2
program stream map version
5
7
program stream info length
16
N loop descriptors
elementary stream map length
16
N loop
CRC 32
32
stream type
8
elementary stream id
8
elementary stream info length
16
N loop descriptors

FIG. 22A

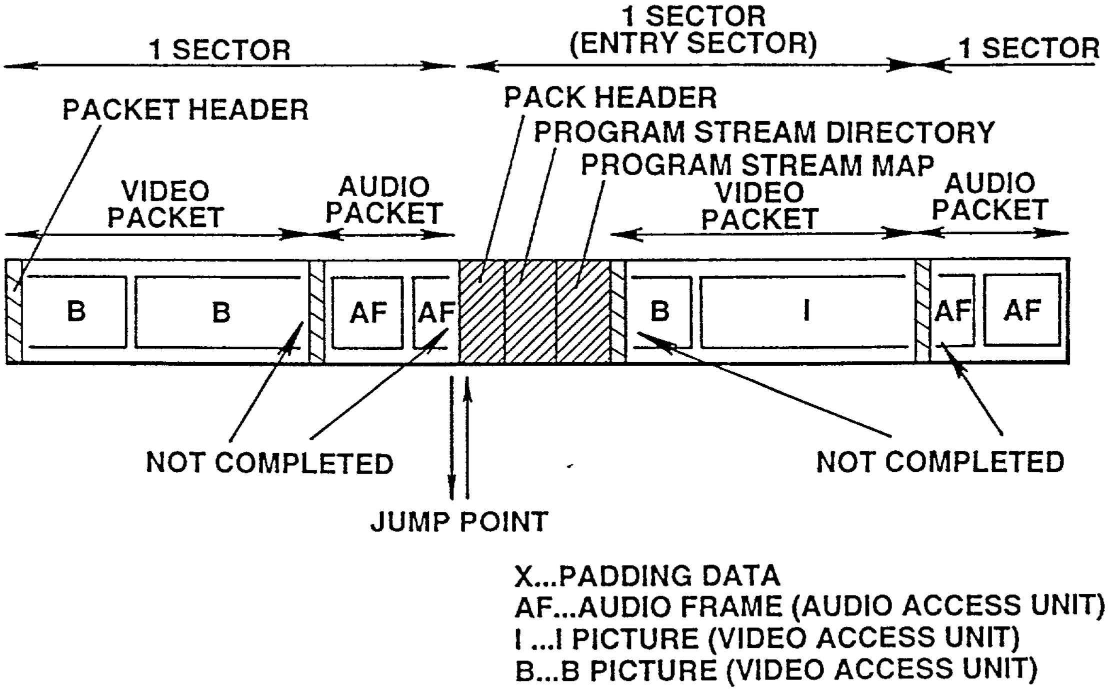
Syntax	No. of Bits	Mnemonic
program_stream_map(){		
packet_start_code_prefix	24	bslbf
map_stream_id	8	uimsbf
program_stream_map_length	16	uimsbf
current_next_indicator	1	bslbf
reserved	2	bslbf
program_stream_map_version	5	uimsbf
reserved	7	bslbf
marker_bit	1	bslbf
program_stream_info_length	16	uimsbf
for (i=0;i<N;i++){		
descriptor()		
}		
elementary_stream_map_length	16	uimsbf
for (i=0;i<N1;i++){		
stream_type	8	uimsbf
elementary_stream_id	8	uimsbf
elementary_stream_info_length	16	uimsbf
for (i=0;i<N2;i++){		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

FIG. 22B



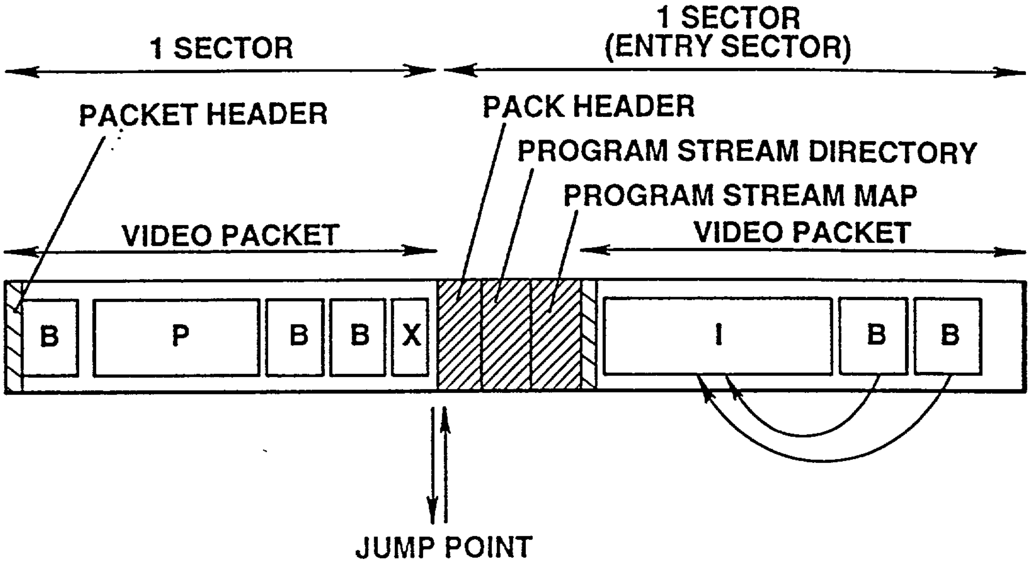
EXAMPLE IN WHICH EACH ACCESS UNIT
OF EACH ELEMENTARY STREAM IS
COMPLETED AT JUMP POINT

FIG.23A



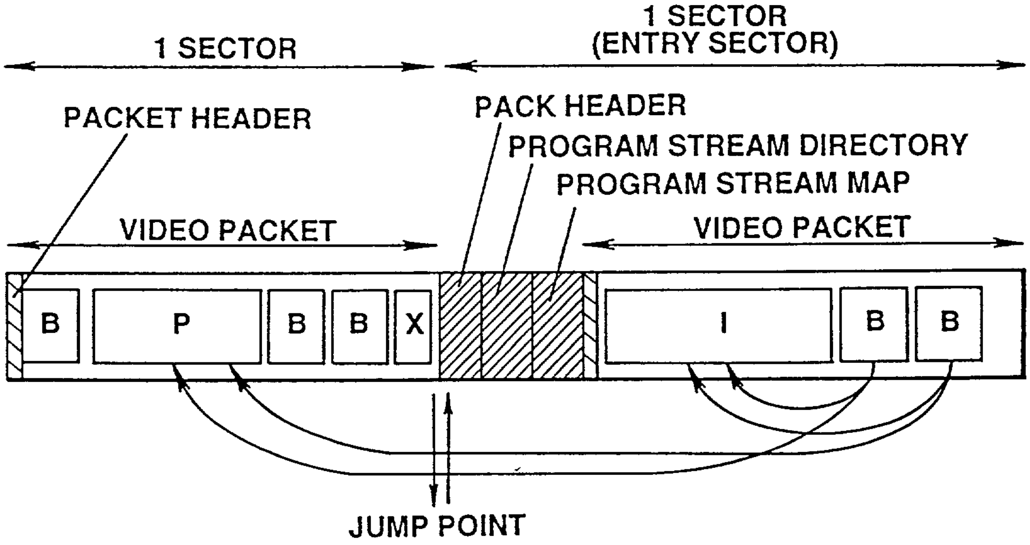
EXAMPLE IN WHICH EACH ACCESS UNIT
OF EACH ELEMENTARY STREAM IS
NOT COMPLETED AT JUMP POINT

FIG.23B



EXAMPLE IN WHICH PREDICTION (REFERENCE) IS NOT MADE ON BOTH SIDES OF JUMP POINT

FIG.24A



X...PADDING DATA
AF...AUDIO FRAME (AUDIO ACCESS UNIT)
I ...I PICTURE (VIDEO ACCESS UNIT)
P...P PICTURE (VIDEO ACCESS UNIT)
B...B PICTURE (VIDEO ACCESS UNIT)

EXAMPLE IN WHICH PREDICTION (REFERENCE) IS MADE ON BOTH SIDES OF JUMP POINT

FIG.24B

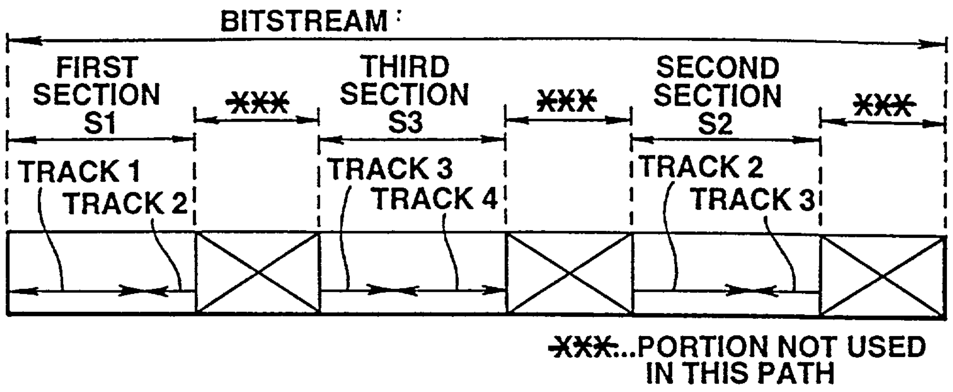


FIG.25

FIG.26A

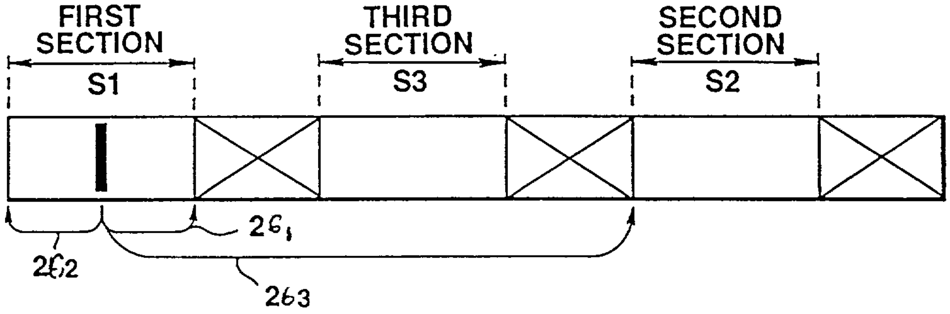


FIG.26B

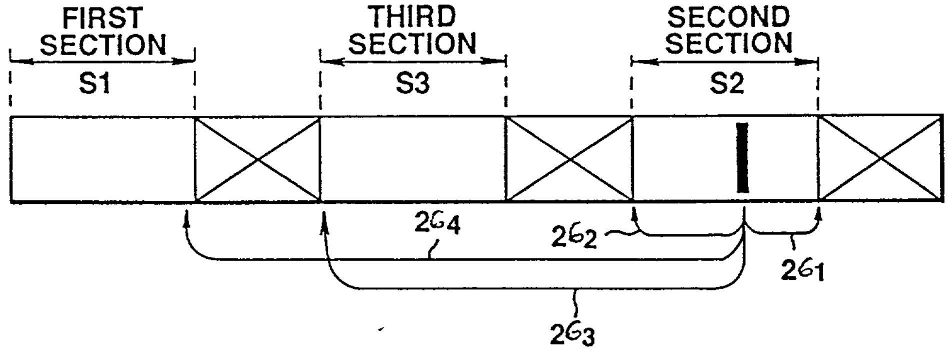


FIG.26C

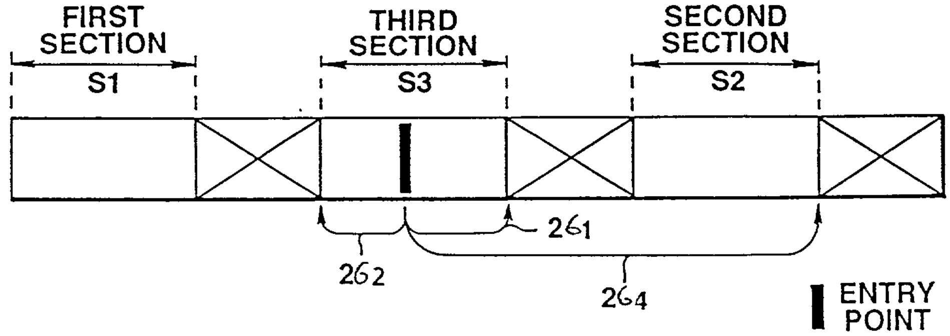


FIG. 27A

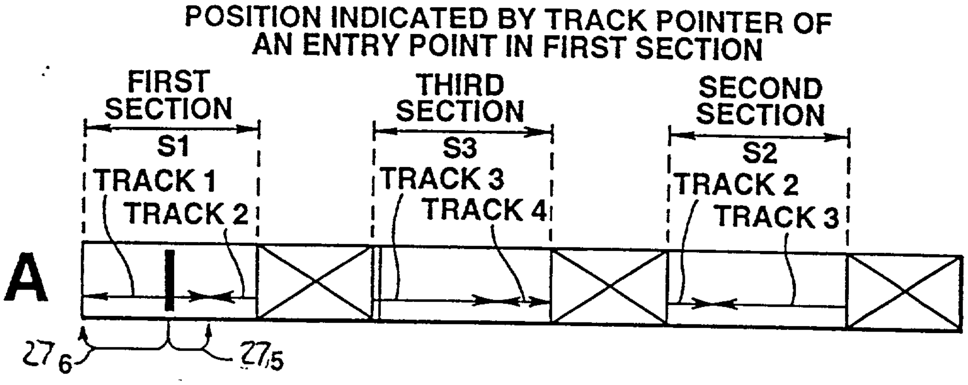


FIG. 27B

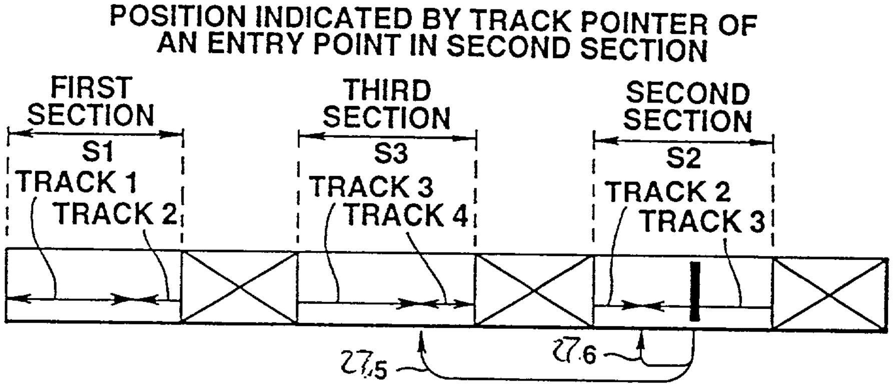
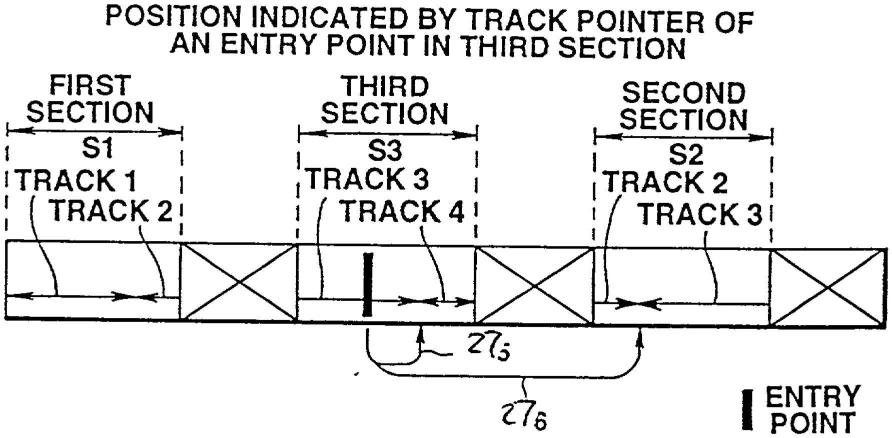


FIG. 27C


POSITION INDICATED BY TRACK POINTER OF AN ENTRY POINT IN THIRD SECTION
FIRST SECTION
S1
TRACK 1
TRACK 2
THIRD SECTION
S3
TRACK 3
TRACK 4
SECOND SECTION
S2
TRACK 2
TRACK 3
27₅
27₆
ENTRY POINT